Lecture 6

TTIC 41000: Algorithms for Massive Data

Toyota Technological Institute at Chicago

Spring 2021

Instructor: Sepideh Mahabadi

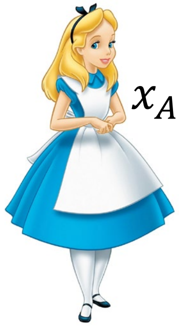
This Lecture

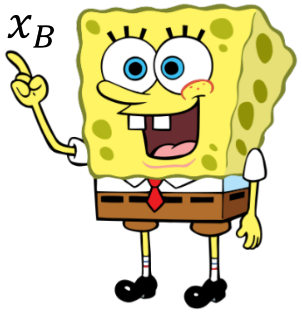
- Proving Lower Bounds in the streaming model
  - Communication Complexity
  - Index Problem + example
  - Set Disjointness + example
  - Gap Hamming Problem + example
  - Set Cover Lower Bounds (if enough time)

Communication Complexity Model

Model

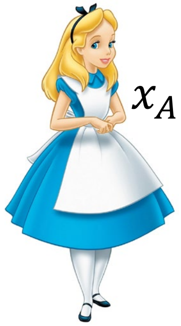
- Two people Alice and Bob, each of them gets an input, e.g., $x_A, x_B \in \{0,1\}^n$
- The goal is to compute a function $f(x_A, x_B)$
- What is the minimum communication required between Alice and Bob

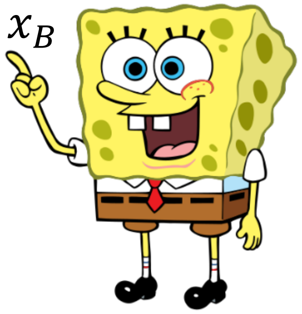




Model

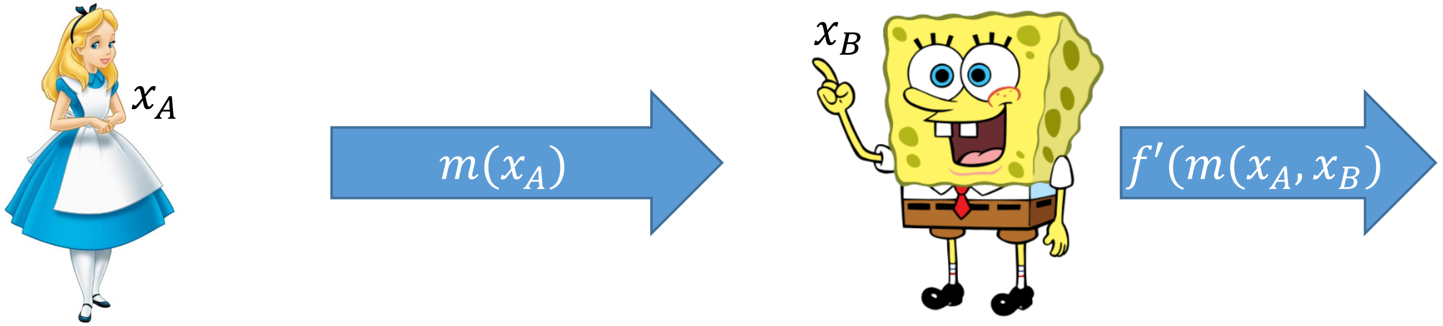
- Two people Alice and Bob, each of them gets an input, e.g., $x_A, x_B \in \{0,1\}^n$, edges of a graph are partitioned between Alice and Bob
- The goal is to compute a function $f(x_A, x_B)$, compute MST
- What is the minimum communication required between Alice and Bob (how many bits)





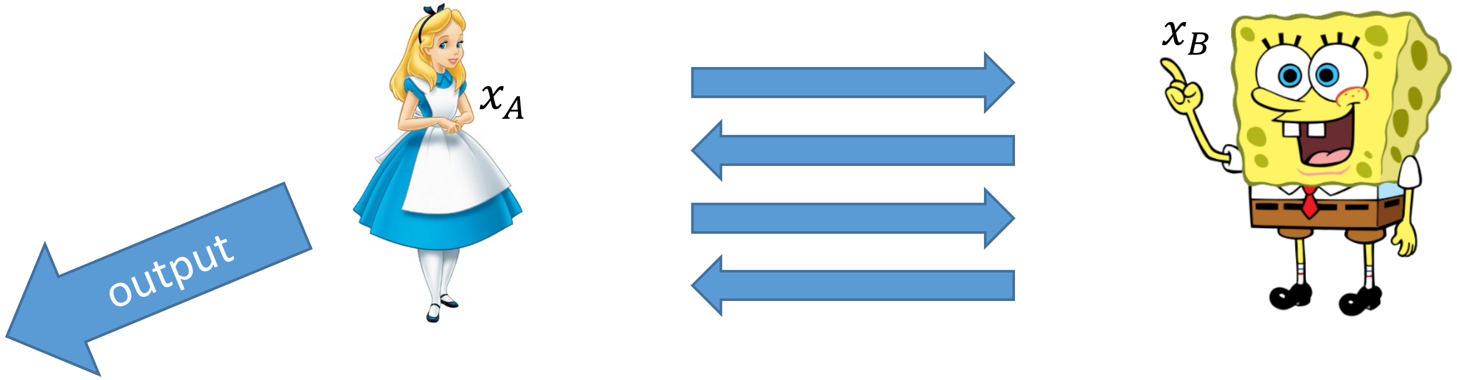
Model

- Two people Alice and Bob, each of them gets an input, e.g., $x_A, x_B \in \{0,1\}^n$
- The goal is to compute a function $f(x_A, x_B)$
- What is the minimum communication required between Alice and Bob (how many bits)
  - One round



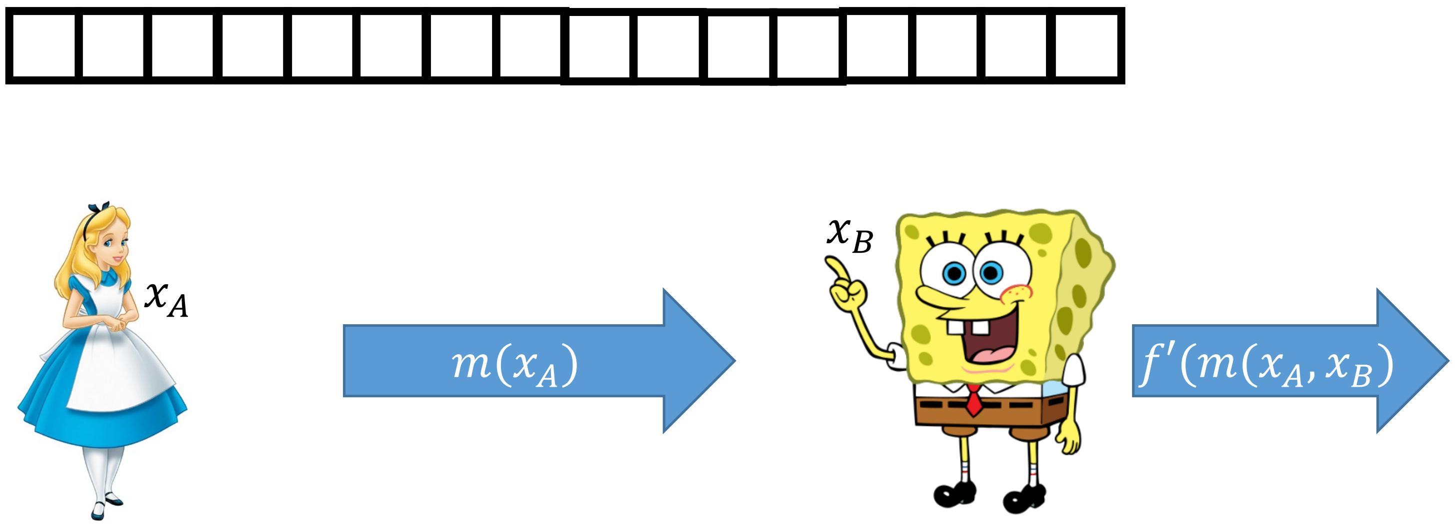
Model

- Two people Alice and Bob, each of them gets an input, e.g., $x_A, x_B \in \{0,1\}^n$
- The goal is to compute a function $f(x_A, x_B)$
- What is the minimum communication required between Alice and Bob (how many bits)
  - One round
  - Multiple rounds



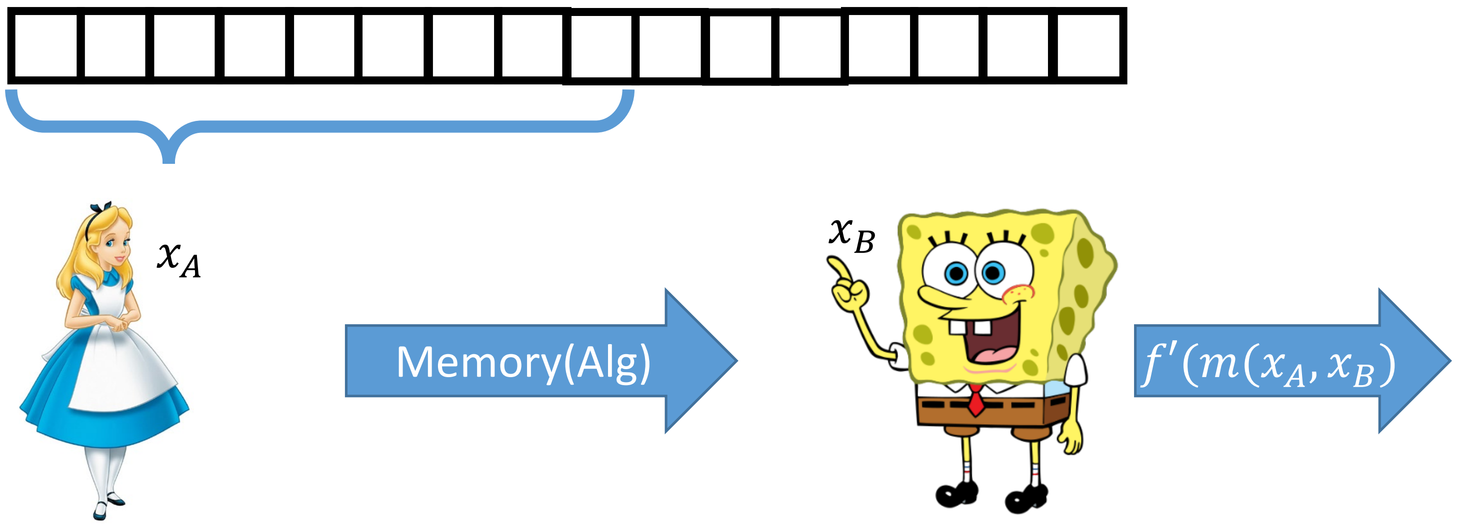
Communication Complexity

- Streaming algorithm -> Communication Complexity Protocol



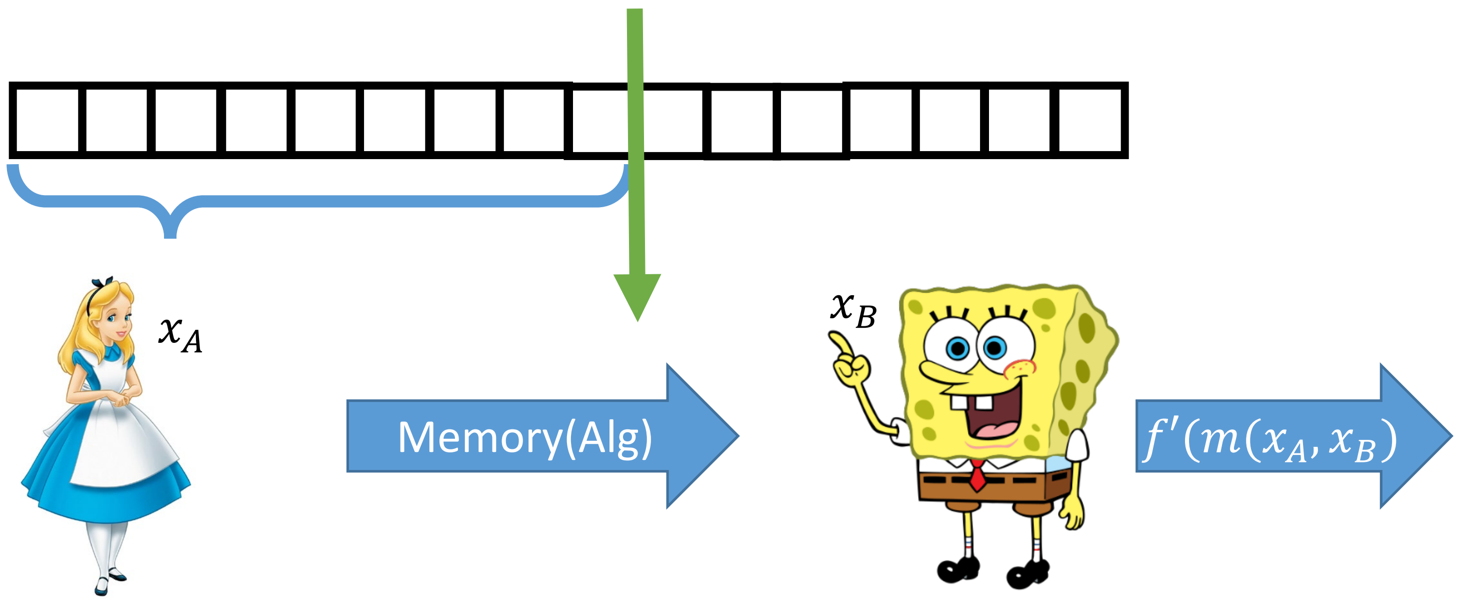
Communication Complexity

- Streaming algorithm -> Communication Complexity Protocol



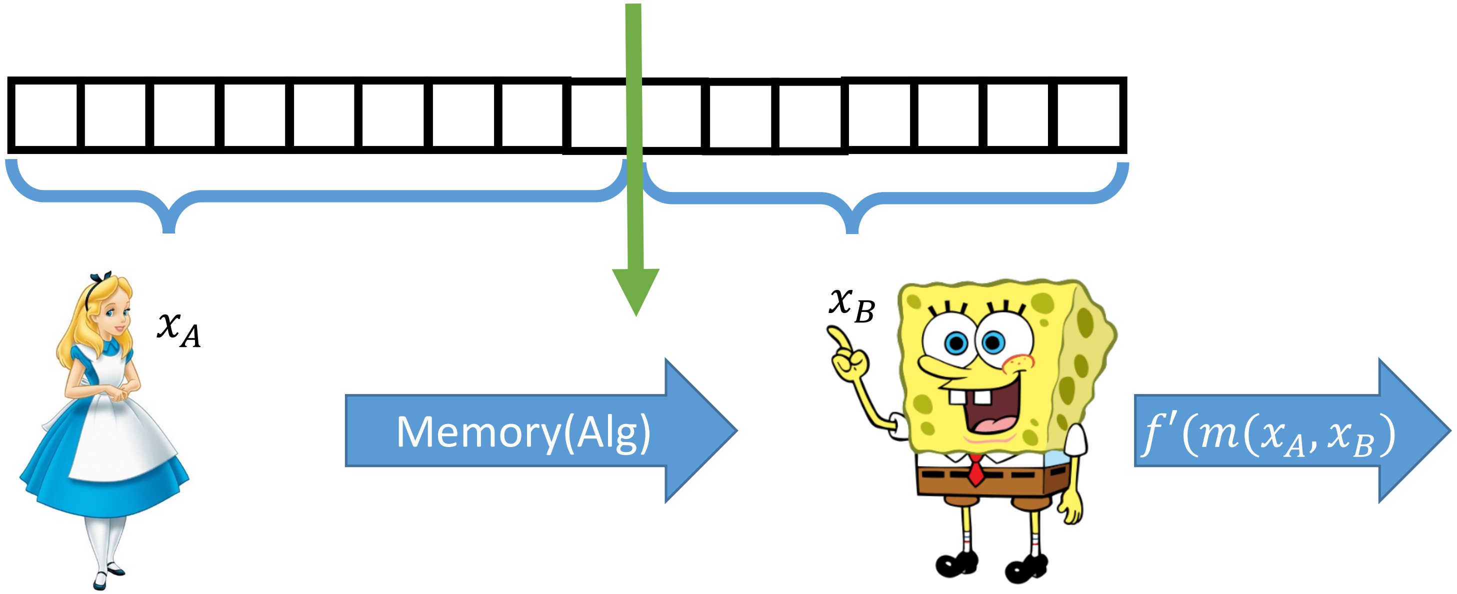
Communication Complexity

- Streaming algorithm -> Communication Complexity Protocol



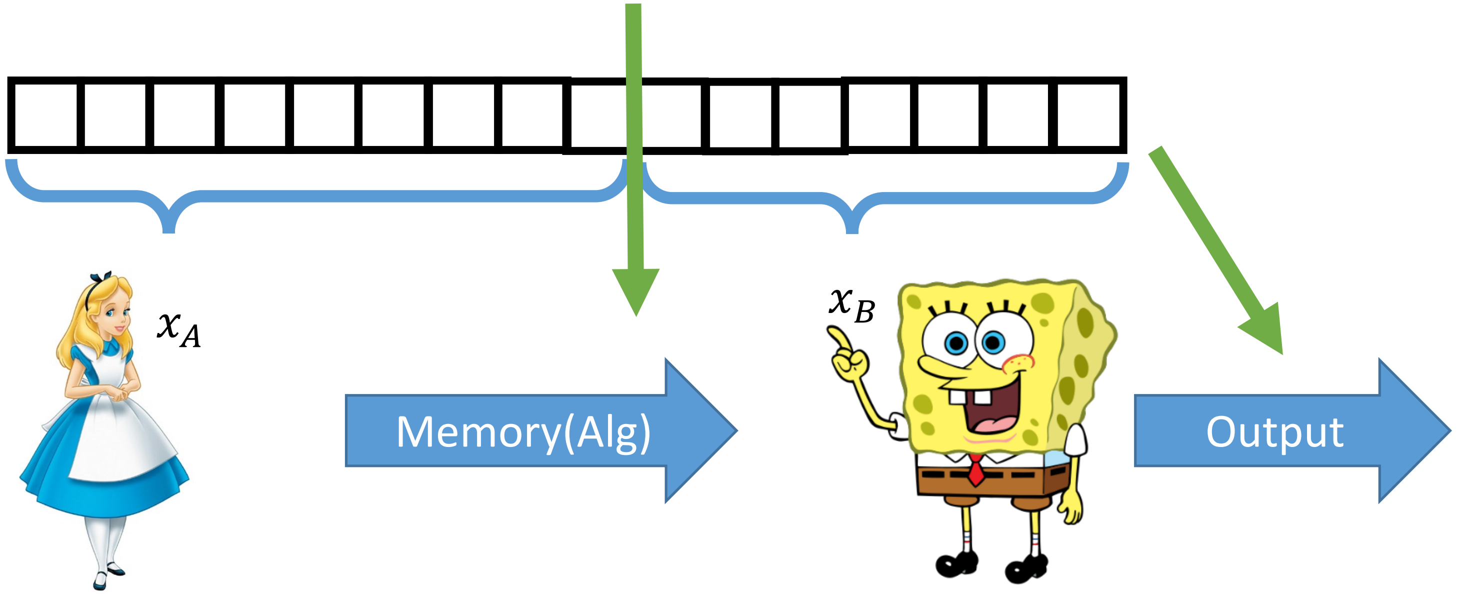
Communication Complexity

- Streaming algorithm -> Communication Complexity Protocol



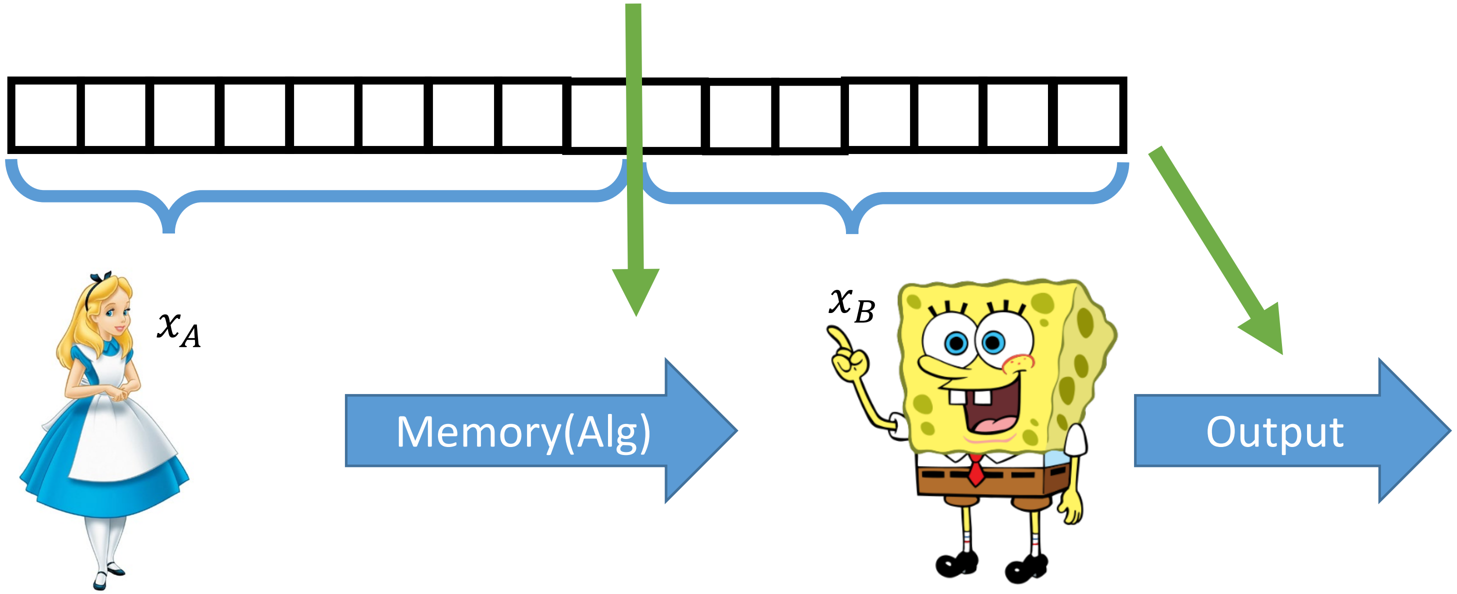
Communication Complexity

- Streaming algorithm -> Communication Complexity Protocol



Communication Complexity

- Single pass streaming algorithm with memory usage s, gives a one round Communication Complexity Protocol with total communication s



Communication Complexity

- Single pass streaming algorithm with memory usage s, gives a one round Communication Complexity Protocol with total communication s
  - p-pass streaming with s bits of space yields a protocol with total cc $(2p-1)s$

Communication Complexity

- Single pass streaming algorithm with memory usage s, gives a one round Communication Complexity Protocol with total communication s
  - p-pass streaming with s bits of space yields a protocol with total cc $(2p - 1)s$

- Any lower bound on the total communication in the CC model, leads to a lower bound on the space usage of any streaming algorithm for the same problem
  - $\Omega(s)$ LB of CC in $(2p - 1)$ rounds $\rightarrow \Omega(\frac{s}{p})$ LB on space of p-pass streaming

Communication Complexity

- Communication Cost of a protocol : worst case (over all possible inputs) number of bits required to transmit
- Communication Complexity: Best possible (over all protocols) Communication cost one can achieve

- Multi party communication with t players
  - Streaming algorithm with s bits space, yields a protocol with total communication $s(t-1)$
- Lower bound of *one-round* multi party with t *players*
  - $\Omega(L)$ LB for total communication, implies $\Omega(L/t)$ LB on space complexity of streaming algorithms
- Randomized Communication Complexity
  - Randomized protocol with public randomness, constant success probability
- Distributional Communication Complexity
  - Inputs of interests are sampled from a given distribution μ

This Lecture

- Proving Lower Bounds in the streaming model
  - Communication Complexity
  - Index Problem + example
  - Set Disjointness + example
  - Gap Hamming Problem + example
  - Set Cover Lower Bounds (if enough time)

The Index Problem

- Alice has a sequence of n bits $x \in \{0,1\}^n$
- Bob has an index $i \in [n]$
- Goal is to output $x(i)$

- One-way (det.) communication complexity of index problem is $\Omega(n)$
  - Suppose Alice sends less than n bits.
  - Then there are two different strings $x_1, x_2 \in \{0,1\}^n$ for which the message from Alice to Bob is the same.
  - If bob queries the bit which is different in x_1 and x_2, he receives the same answer which is a contradiction.

The Index Problem

- Alice has a sequence of n bits $x \in \{0,1\}^n$
- Bob has an index $i \in [n]$
- Goal is to output $x(i)$

- One-way (deterministic) communication complexity of index problem is $\Omega(n)$
- One-way (randomized) communication complexity of index problem is $\Omega(n)$

Streaming Lower Bound for Connectivity using the Index Problem

Given *edges* of a graph in the streaming fashion, decide if it is connected.

Reduction from Index Problem

- Alice has a sequence of n bits $x \in \{0,1\}^n$
- She builds a graph with a node $s \cup \{v_1, \dots, v_n\}$ where $(s, v_i) \in E$ iff $x_i = 1$.
- Bob has an index $i \in [n]$
- He adds a vertex t and connect it to all v_j but v_i and connects it to s
- Checking connectivity implies knowing x_i

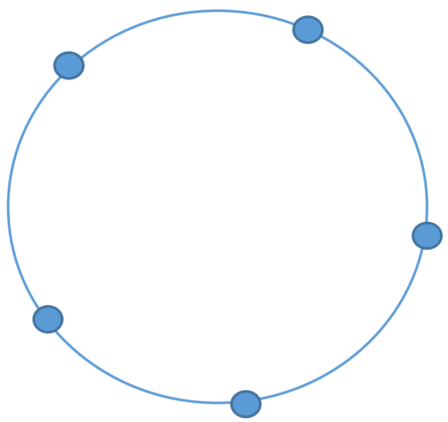
2dim SVM Lower Bound

Index Problem: Alice has m bits. Bob has an index i. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Instance

- m locations on a circle corresponding to Alice's bits
- n/m points on each location if the corresponding bit is 1, otherwise no point



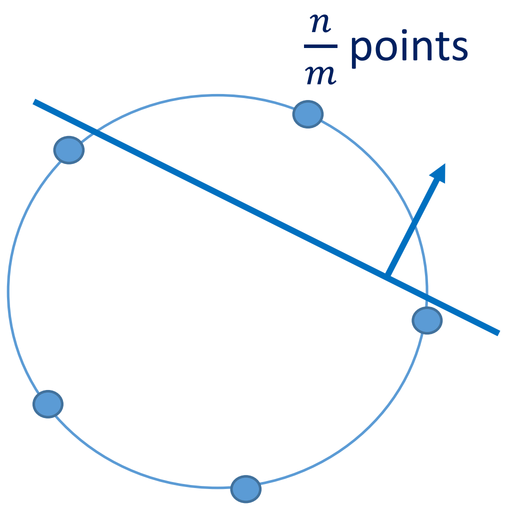
2dim SVM Lower Bound

Index Problem: Alice has $\boldsymbol{m}$ bits. Bob has an index $\boldsymbol{i}$. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Instance

- $\boldsymbol{m}$ locations on a circle corresponding to Alice's bits
- $\boldsymbol{n/m}$ points on each location if the corresponding bit is 1, otherwise no point
- Bob can query the hyperplane excluding i-th point to find out Alice's bit



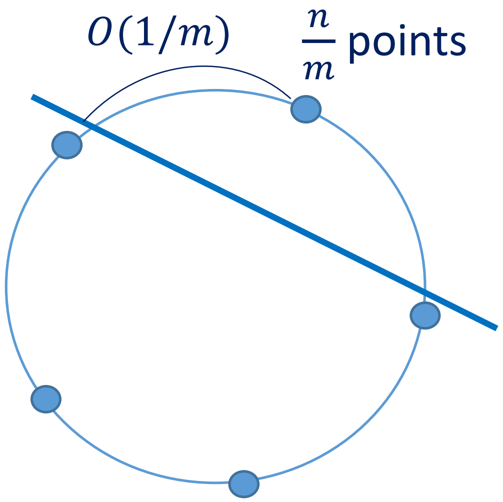
2dim SVM Lower Bound

Index Problem: Alice has m bits. Bob has an index i. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Instance

- m locations on a circle corresponding to Alice's bits
- n/m points on each location if the corresponding bit is 1, otherwise no point
- Bob can query the hyperplane excluding i-th point to find out Alice's bit



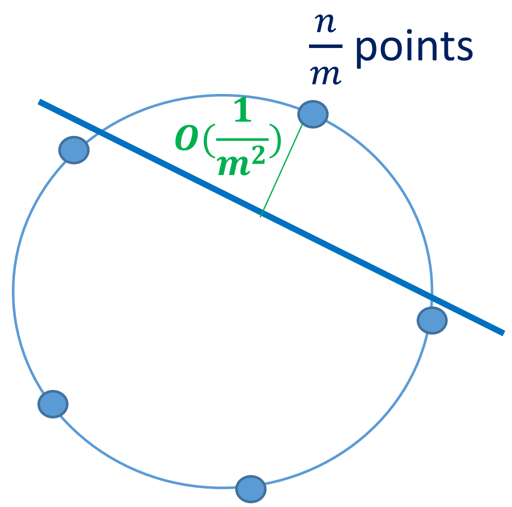
2dim SVM Lower Bound

Index Problem: Alice has m bits. Bob has an index i. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Instance

- m locations on a circle corresponding to Alice's bits
- n/m points on each location if the corresponding bit is 1, otherwise no point
- Bob can query the hyperplane excluding i-th point to find out Alice's bit
- Total additive error is $O(\frac{n}{m} \cdot \frac{1}{m^2})$



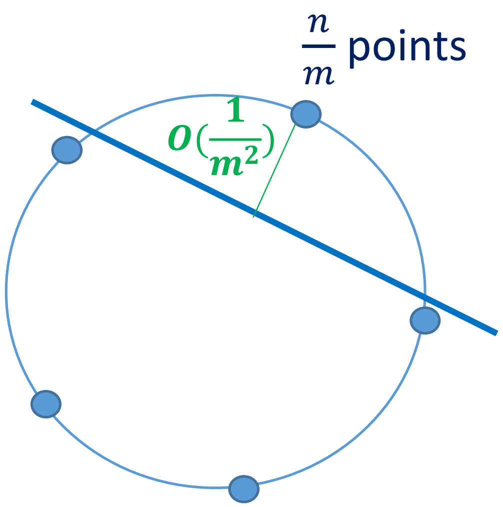
2dim SVM Lower Bound

Index Problem: Alice has m bits. Bob has an index i. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Instance

- m locations on a circle corresponding to Alice's bits
- n/m points on each location if the corresponding bit is 1, otherwise no point
- Bob can query the hyperplane excluding i-th point to find out Alice's bit
- Total additive error is $O(\frac{n}{m} \cdot \frac{1}{m^2})$
- Thus achieving error $O(n\epsilon)$ requires space $\Omega(\epsilon^{-\frac{1}{3}})$



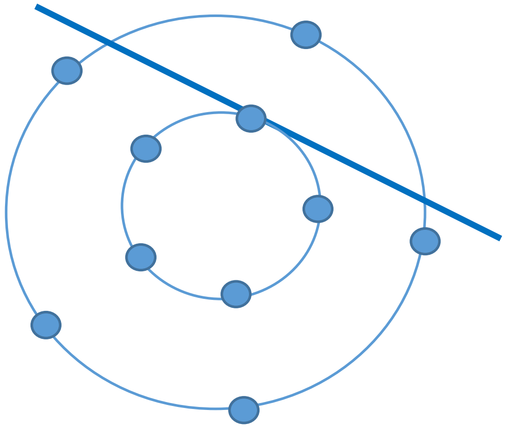
2dim SVM Lower Bound

Index Problem: Alice has $\boldsymbol{m}$ bits. Bob has an index $\boldsymbol{i}$. Bob wants to know whether the ith bit of Alice is 0 or 1.

- This requires $\Omega(m)$ space

Improvement

- Peel the instance and figure out all the bits
- This allows us to encode more bits
- This improves the lower bound to $\Omega(\epsilon^{-\frac{3}{5}})$



Index Problem

- It only works for one way protocols
- Bob can easily send $O(\log n)$ bits to Alice
- Does not work for a general communication protocol (only one-way)

This Lecture

- Proving Lower Bounds in the streaming model
  - Communication Complexity
  - Index Problem + example
  - Set Disjointness + example
  - Gap Hamming Problem + example
  - Set Cover Lower Bounds (if enough time)

Set Disjointness

- Alice and Bob each have a bit string $x_A, x_B \in \{0,1\}^n$
- Goal: Decide if they are disjoint or not, i.e., $\exists i: x_A(i) = x_B(i) = 1$
- Total communication required between Alice and Bob is $\Omega(n)$

Multi Party Set Disjointness

- There are t parties
- Each hold a bit string $\in \{0,1\}^n$
- For each index i, there is either no party, one party, or all parties with that bit equal to 1.

	1	2	3	4	5	6
player 1	0	0	0	1	0	0
player 2	1	0	0	1	1	0
player 3	0	1	0	1	0	0
player 4	0	0	0	1	0	0

Question: Is there an index i included by all parties?

- Total Communication required between all parties is $\Omega(n/t)$

Frequency Moment Problem

- Given a stream S of items $i_1, \cdots, i_m$ where each item belongs to $[n]$

k-th moment of S is defined as

$$F_k(S) = x_1^k + \cdots + x_n^k,$$

where x_j is the number of times j appears in the stream S

Streaming Frequency Moment LB using Set Disjointness

- Reduction from t-party set disjointness with bit string of length n
- Each player i generates S_i, a set of indices contained by i
- The final stream is $S = S_1, \dots, S_t$
  - $\{4, \quad 1, 4, 5, \quad 2, 4, \quad 4\}$

	1	2	3	4	5	6
player 1	0	0	0	1	0	0
player 2	1	0	0	1	1	0
player 3	0	1	0	1	0	0
player 4	0	0	0	1	0	0

- **Claim.** If all indices are contained by 0 or 1 party, then $F_k(S) \le n$.

  Proof. $F_k(S) = \left(x_1^k + x_2^k + \cdots + x_n^k\right) \le n$ *(x_i denotes the number of times index i appears in S; $\forall i, x_i \in \{0,1\}$)*
- **Claim.** If an index contained by all parties, then $F_k(S) \ge t^k$.

  Proof. $F_k(S) = \left(x_1^k + x_2^k + \cdots + x_n^k\right) \ge t^k$ *($\exists i, x_i = t$)*

→ if $t^k > 2n$, then a 2-approx. of F_k solves t-party set disjointness with bit string of length n

An s-space streaming 2-approximation of F_k $\xrightarrow{\text{yields}}$ $s(t-1)$ *bit protocol of t*-party set disjointness with bit string of length n

$\Omega(n/t)$ CC of t-party set disjointness(n) → 2-approximation streaming alg. of F_k requires $\Omega\left(\frac{n}{t^2}\right) = \Omega(n^{1-\frac{2}{k}})$ bits space

This Lecture

- ❑ Proving Lower Bounds in the streaming model
  - Communication Complexity
  - Index Problem + example
  - Set Disjointness + example
  - Gap Hamming Problem + example
  - Set Cover Lower Bounds (if enough time)

Gap Hamming Problem

- Alice and Bob each have a bit string $x_A, x_B \in \{0,1\}^n$
- Goal: Compute the Hamming distance between x_A and x_B

- Computing Hamming distance within an additive error of $\sqrt{n}$ requires $\Omega(n)$ communication (e.g., deciding if $H(x_A, x_B) \geq \frac{n}{2} + \sqrt{n}$ or $H(x_A, x_B) \leq \frac{n}{2} - \sqrt{n}$)

Streaming LB for Distinct Elements using Gap Hamming

- Reduction from Hamming distance between x_A and x_B
  - S_A: the indices of 1-bit in Alice's string
  - S_B: the indices of 1-bit in Bob's string

Observation. $2F_0(S) = |x_A| + |x_B| + \Delta(x_A, x_B)$

Hamming distance is hard even if we know both x_A and x_A have exactly $(\frac{n}{2})$ 1s.

Claim. $(1+\epsilon)$-approx. of DE, approximate Hamming distance within

$$\frac{\epsilon\left(n + \Delta(x_A + x_B)\right)}{2} \leq n\epsilon$$

Hence, for $\epsilon \approx 1/\sqrt{n}$, any $(1+\epsilon)$-approx. of DE has space complexity $\Omega(1/\epsilon^2)$

This Lecture

- ❑ Proving Lower Bounds in the streaming model
  - Communication Complexity
  - Index Problem + example
  - Set Disjointness + example
  - Gap Hamming Problem + example
  - Set Cover Lower Bounds (if enough time)

Lower bound: single pass

- Have seen that $O(1)$ passes can reduce space requirements
- What can(not) be done in one pass?
- We show that distinguishing between $k = 2$ and $k = 3$ requires $\widetilde{\Omega}\ (mn)$ space

Many vs One Set-Disjointness

- Two sets cover U iff their complements are disjoint
- Consider the following one-way communication complexity problem:
  - Alice: sets $S_1, \ldots, S_m$
  - Bob: set S_B
  - Question: is S_B disjoint from any of S_i's ?

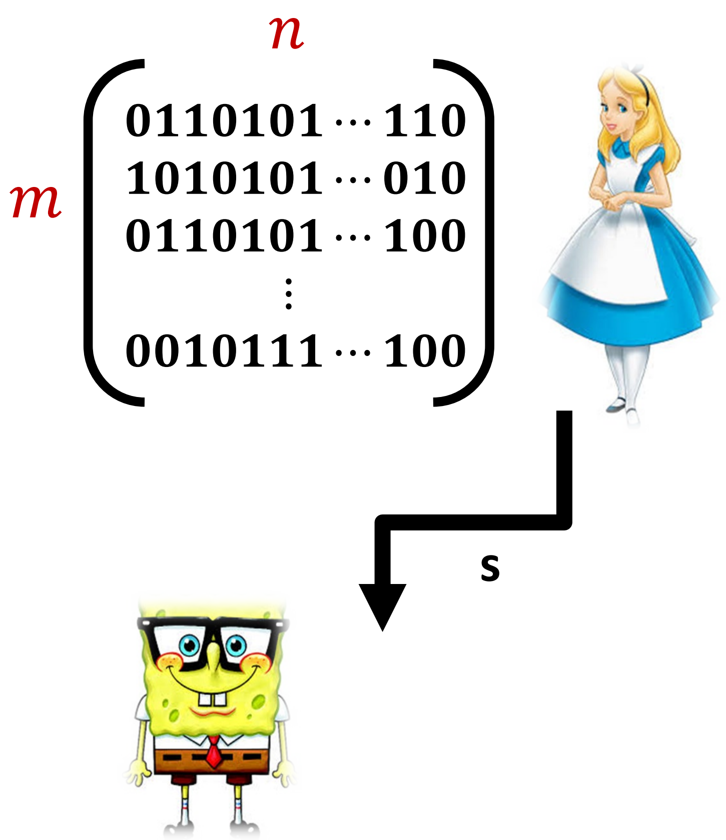
The randomized one-round communication complexity of Many vs. One Set-Disjointness is $\Omega(mn)$ if error probability is 1/poly(m).

Many vs One Set-Disjointness

The randomized one-round communication complexity of Many vs. One Set-Disjointness is $\Omega(mn)$ if error probability is 1/poly(m).

- Alice's sets are selected *uniformly* at random
- There exist poly(m) sets S_B such that if Bob learns answers to all of them, he can recover all S_i's with high probability

- Bob can recover mn random bits from $o(mn)$ bits of communication -> contradiction



Recovering Alice's Collection

- Recovery procedure
  - Suppose that Bob has a set S_B that is disjoint from *exactly* one S_i (we do not know which one)
    - Call it a "good seed" for S_i
  - Then Bob queries all extensions $S_B \cup \{e\}$ to recover S_i
- Bob's queries:
  - A random "seed" of size $c \log m$ is disjoint from exactly one S_i w.p. $m^{-O(c)}$
  - Try $m^{O(c)}$ times
- Recover all S_i